Cross-validation: What does it estimate and how well does it do it?

by Stephen Bates, Trevor Hastie, and Robert Tibshirani

Gül İnan

Istanbul Technical University

Machine Learning in Montpellier, Theory & Practice,
January 6th, 2022

Motivation

- In machine learning applications, when deploying a **predictive model**, the main interest is on understanding **prediction accuracy** of the model on future test points.
- The standard measure of accuracy for a predictive model is the **prediction error**, i.e., the **expected loss on future test points**.
- For inference, both **good point estimates** and **accurate confidence intervals** are required for **prediction error**.

Motivation continu'ed

- From a practical point of view, **cross-validation (CV)** is one of the widely-used resampling-based approaches to estimate a **point estimate** and to build a **confidence interval** for prediction error.
- However, **Bates, Hastie, and Tibshirani (2021)** discuss that (statistical) properties of CV estimator of prediction error are **not well-understood** despite CV has a very simple functionality.

Aim

- **Bates, Hastie, and Tibshirani (2021)** firstly show that the **CV estimator of prediction error**:
  - tracks **the accuracy of the model fit weakly** and, instead
  - estimates **the average prediction error of models fit across many (hypothetical) data sets from the same population**.

Aim continu'ed

- **Bates, Hastie, and Tibshirani (2021)** secondly show that the **naive confidence intervals** based on CV estimate of prediction error give **poor coverage** since the variance of error estimates used to compute the width of the interval does not account for the **correlation between error estimates** in different folds, which arises from the fact that each data point is used both in training and testing.
- **Bates, Hastie, and Tibshirani (2021)** propose the **nested cross-validation (NCV)** approach which provides confidence intervals with a **coverage close to the nominal level**.
- **Bates, Hastie, and Tibshirani (2021)** validate their work through deep theory and extensive numerical experiments (both simulation studies and real data examples).

Setting and notation

- Consider a **supervised learning** setting.
- We have a data (X, Y), where $X = (X_1, \ldots, X_n) \in \mathcal{X}^{n \times p}$ is the **feature matrix** and $Y = (Y_1, \ldots, Y_n) \in \mathcal{Y}^n$ is the **response vector**.
- We assume that each **data point** (X_i, Y_i), $i = 1, \ldots, n$, is i.i.d. from a **distribution** P.

Setting and notation continu'ed

- Consider a **class of models** parameterized by vector θ.
- We assume that $\widehat{f}(x, \theta)$ is the **function that predicts** y from $x \in \mathbb{R}^p$ using the model with parameter θ , where θ takes values in some space Θ.
- We let $\mathcal{A}$ be a **model-fitting algorithm** that returns the **fitted value** of the parameter vector, $\hat{\theta} = \mathcal{A}(X, Y) \in \Theta$ based on the observed data (x, y).

Prediction error

- In **measuring the accuracy of a model**, we are interested in **prediction error (out-of-sample error)** which is defined as the **expected loss on future data points** (X_{n+1}, Y_{n+1}):

$$Err_{XY} := \mathbb{E}[\ell\big(\hat{f}\,(X_{n+1}, \hat{\theta}), Y_{n+1}\big)|(X, Y)],$$

- where (X_{n+1}, Y_{n+1}) is an **independent test point** from the same distribution P.
- The expression $\hat{f}\,(X_{n+1}, \hat{\theta})$ is the **predicted value** of Y_{n+1} at the future point X_{n+1} and
- $\hat{\theta}$ is the fitted value of the parameter estimated through algorithm $\mathcal{A}$ based on the training data (X, Y).
- The expression $\ell\big(\hat{f}\,(X_{n+1}, \hat{\theta}), Y_{n+1}\big)$ is the **loss** between predicted value of Y_{n+1} and Y_{n+1} itself.
- Here, the **loss function** $\ell(.\,)$ could be squared error loss, classification error, or deviance (cross-entropy).
- Furthermore Err_{XY} can be considered as a **random quantity** depending on the training data (X, Y).

Expected prediction error

- On the other hand, we may also be interested in learning algorithms' **average performance on predicting future test points** when designing and comparing **algorithms** with each other.
- This quantity of interest can be formally defined as the **expected value of prediction error** across possible training data sets of size n drawn from the same data distribution P:

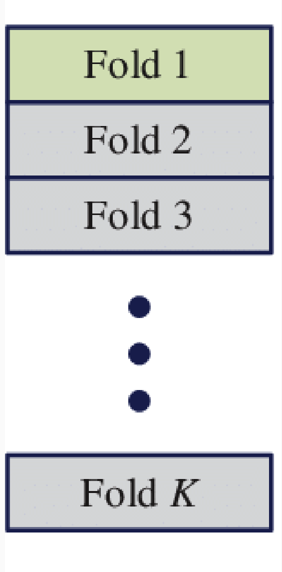
$$Err := \mathbb{E}\left[Err_{XY}\right].$$

- Shortly: it is the expectation of prediction error across possible training sets (from the same distribution).

- Note that estimates of the quantities Err_{XY} and Err **cannot be computed** when the **data distribution** P is **unknown**.
- Then, **resampling based methods** such as cross-validation, bootstrap, and jacknife or **analytical methods** such as AIC, BIC, Mallow's C_p, and covariance penalties can be used to **estimate the quantities** Err_{XY} and Err.

K-fold cross-validation

- In **K-fold cross-validation**, we randomly partition the data $(X, Y) = \mathcal{I}$ into K equally sized **disjoint folds (subsets)** $\mathcal{I}_k$ $(k = 1, \dots, K)$.
- Here the fold size is $m = n/K$ and the whole data is $\mathcal{I} = \cup_{k=1}^{K} \mathcal{I}_k$.
- When the data point $(x_i, y_i) \in \mathcal{I}_k$, we will also write $i \in \mathcal{I}_k$ $(k = 1, \dots, K)$.

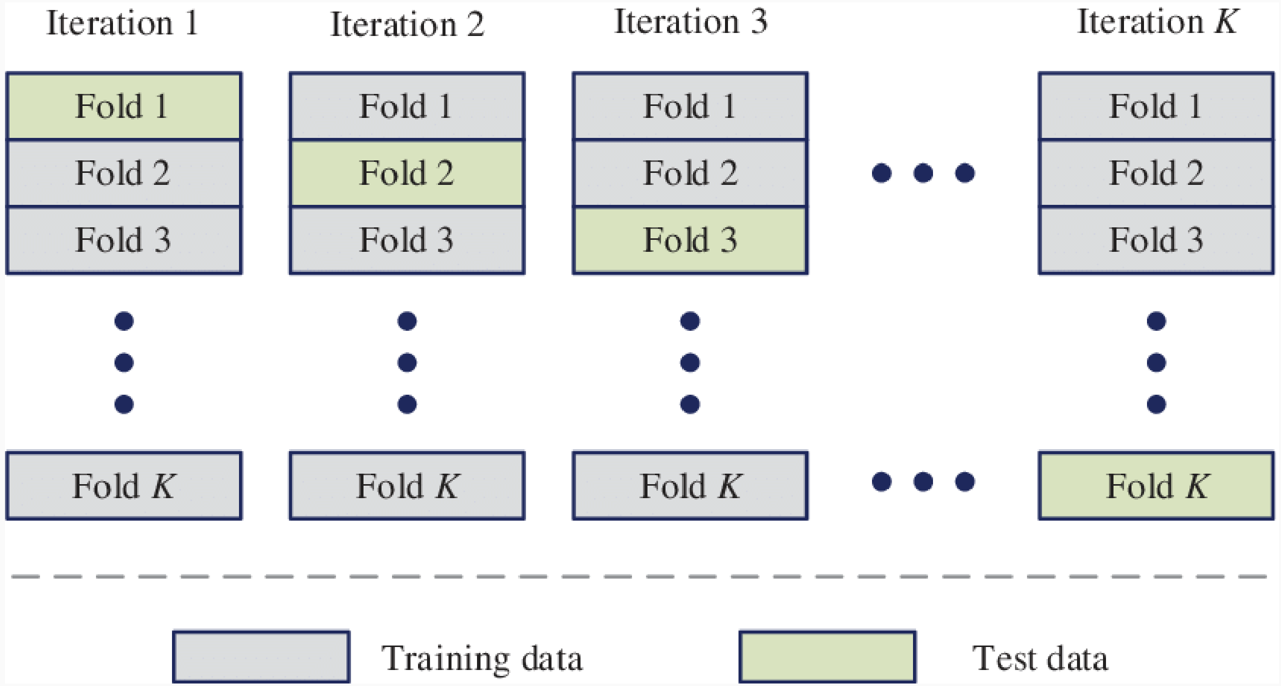


Partition of the data into K-folds.

- Consider the first fold $\mathcal{I}_1$ and hold it out as future data set (or **test set**).
- The **remaining data points** $(x_i, y_i) \in \mathcal{I} \backslash \mathcal{I}_1$, which are not in the first fold, are called as **training set**.
- Let $\hat{\theta}^{(-1)} = \mathcal{A}\big((X_i, Y_i)_{i \in \mathcal{I} \backslash \mathcal{I}_1}\big)$ be the **parameter estimate based on training data set**, then we can calculate the **prediction error** for future data set $\mathcal{I}_1$, of size m, as follows:

$$\frac{1}{m} \sum_{i \in \mathcal{I}_1} \ell\big(\hat{f}(x_i, \hat{\theta}^{(-1)}), y_i\big).$$

- In **K-fold cross-validation**, we **iteratively repeat** this process for each fold $(k = 1, \ldots, K)$.



K-fold cross-validation.

Image Source

CV estimate of prediction error

- The **average of prediction errors over K folds** is given as follows:

$$\hat{Err}^{(CV)} := \frac{1}{K}\sum_{k=1}^{K}\frac{1}{m}\sum_{i\in\mathcal{I}_k}\ell\big(\hat{f}\,(x_i,\hat{\theta}^{(-1)}),y_i\big).$$

- This is usually called as the **CV estimate of prediction error**.
- **Relationship between $\hat{Err}^{(CV)}$, Err_{XY}, and Err**: Intuitively, the inner sum is an estimate for Err_{XY} for a fixed fold, and the double sum estimates Err with $\mathcal{I}_k$ $(k = 1, \ldots, K)$ being different samples from the same distribution **(De Benito Delgado, 2021)**.

What prediction error are we estimating?

- Err_{XY} is the **prediction error of the model which is fit on the training data set**.
- Err is the **average of the fitting algorithm runs on the same-sized data sets drawn from the same distribution P.**
- The **former quantity** is of the most interest to a practitioner **deploying a specific model**, whereas the **latter** may be of interest to a researcher **comparing different fitting algorithms**.

- Some earlier studies such as Zhang (1995), Hastie et al. (2009), and Yousef (2020) have observed that **cross-validation estimate provides little information** about Err_{XY}, which is also called as *weak correlation* problem in the literature.
- For the special case of the linear model, **Bates, Hastie, and Tibshirani (2021)** claim that CV estimate should be **considered as an estimate** of Err rather than Err_{XY}.

Err_X: A different target of inference

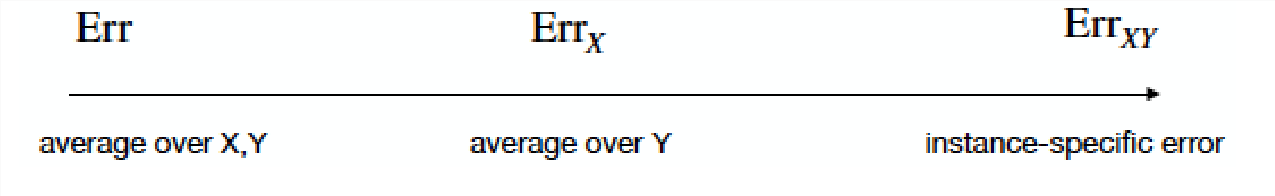
- Assume a **homoskedastic Gaussian linear model** as follows:

$$y_i = x_i^T\theta + \epsilon_i \quad \text{where} \quad \epsilon_i \overset{i.i.d.}{\sim} N(0,\sigma^2) \quad \text{i=1,...,n.}$$

- Define a **new (mid) key quantity** as follows:

$$Err_X := \mathbb{E}[Err_{XY}|X],$$

- which **falls between** Err and Err_{XY} as visualized below:



Possible targets of inference for cross-validation.

- **Lemma 1:** When ordinary least squares (OLS) is used as the fitting algorithm along with a squared-error loss function, the CV estimate of prediction error, $\hat{Err}^{(CV)}$, is linearly invariant.

- **<<** Under this setting, residuals turns out to be the same for both **original** $(x_1, y_1), \ldots, (x_n, y_n)$ and **shifted data** $(x_1, y_1^{'}), \ldots, (x_n, y_n^{'})$, where $(y_i^{'} = y_i + x_i^T \kappa)$. Since the CV estimate of prediction error is the mean of the squared residuals, the CV estimate of prediction error also turns out to be **the same** for both the **original data** and the **shifted data**. **>>**

- **Theorem 1:** Assume homoskedastic Gaussian linear model holds and that we use squared-error loss function. Let $\hat{E}rr$ be a **linearly invariant estimate of prediction error** (such as $\hat{E}rr^{(CV)}$ using OLS as the fitting algorithm). Then,

$$Err_{XY} \perp \hat{E}rr \quad | \quad X.$$

- << Recall from classical linear regression theory that when using OLS, the **estimated coefficient** vector $\hat{\theta}$ is independent of the **residuals** $(Y - X\hat{\theta})$:

$$\hat{\theta} \perp (Y - X\hat{\theta}) \quad | \quad X.$$

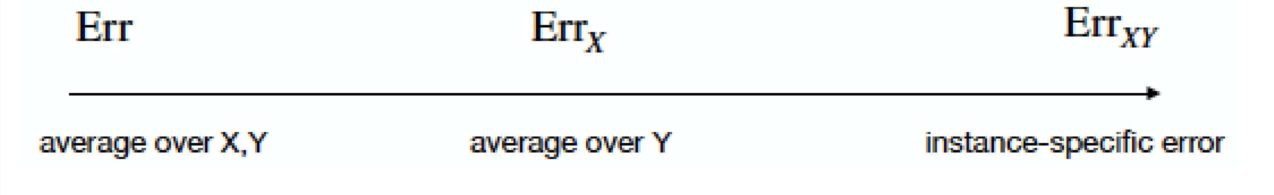
- Since Err_{XY} is a function of $\hat{\theta}$ only, which is the OLS estimate of θ, and any linearly invariant estimate of prediction error Err is a function only of residuals, $Y - X\hat{\theta}$, by the invariance property, then $Err_{XY} \perp \hat{E}rr \quad | \quad X.$ >>

- **Corollary 1:** Under the conditions of Theorem 1, we get the following decomposition:

$$\mathbb{E}[\big(\widehat{E}rr - Err_{XY}\big)^2] = \mathbb{E}[\big(\widehat{E}rr - Err_X\big)^2] + \mathbb{E}[Var\big(Err_{XY}|X\big)].$$

- << Any linearly invariant estimator (such as cross-validation) has **larger mean squared error** (MSE) as an estimate of Err_{XY} than as an estimate of Err_X. >>

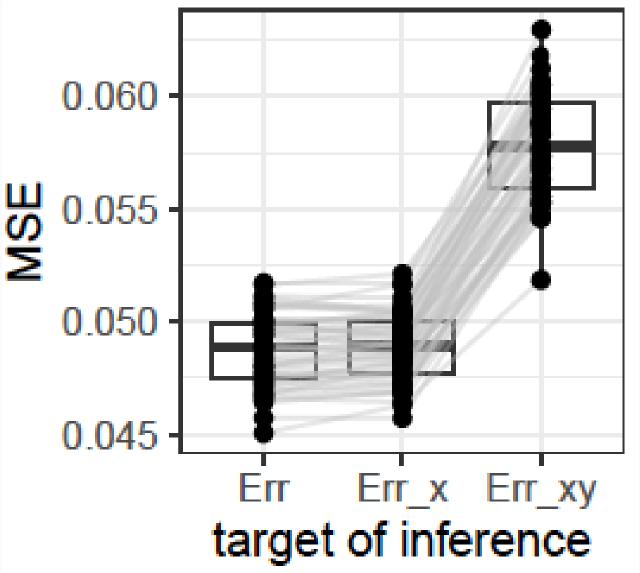
- This implies that $\widehat{E}rr^{CV}$ is a better estimate of the intermediate quantity Err_X than of Err_{XY}.



Possible targets of inference for cross-validation.

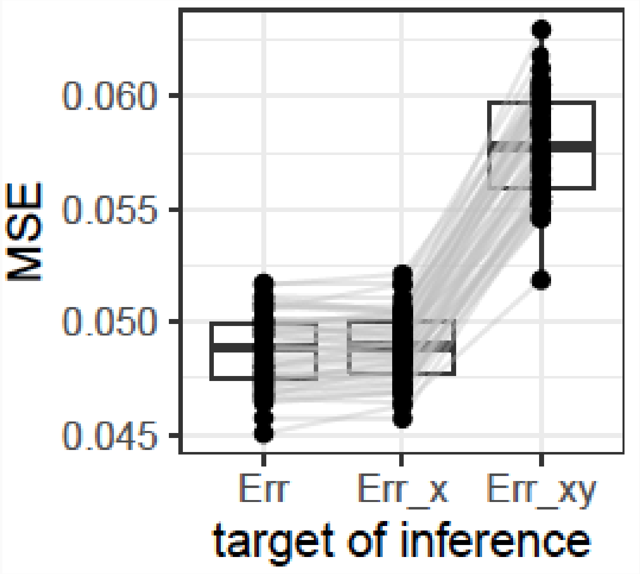
Example

- Consider an experiment in a simple linear model with $n = 100$ observations and $p = 20$ features $\overset{i.i.d.}{\sim} N(0, 1)$, which is replicated 2000 times.
- **MSE** of $\hat{Err}^{(CV)}$ relative to **three estimands**: Err, Err_X, and Err_{XY}.

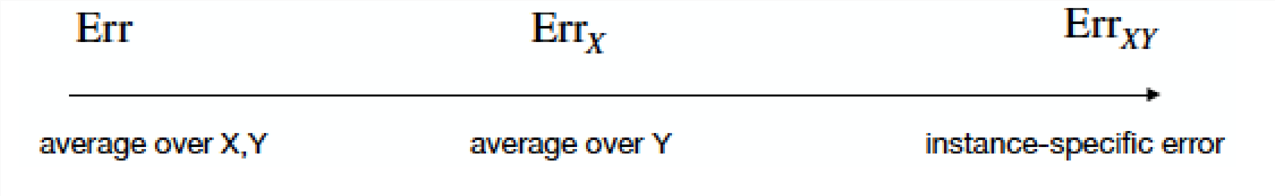


- **Side note:** Each pair of points connected by a line represents the 2000 replicates with the same feature matrix X.

- We see that $\hat{Err}^{(CV)}$ has **lower MSE** for Err_X than Err_{XY}.
- These results suggest that, Err_X is a more **natural target of inference** (estimand) rather than Err_{XY} for $\hat{Err}^{(CV)}$.



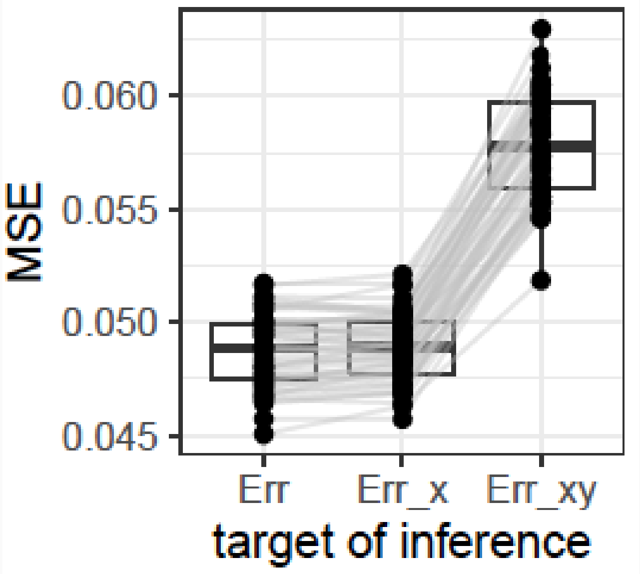
- **Bates, Hastie, and Tibshirani (2021)** further investigate the **relationship** between Err and Err_X through asymptotic analysis.



Possible targets of inference for cross-validation.

- **Bates, Hastie, and Tibshirani (2021)** show that the variance of Err_X (which also has mean Err) is **small** compared with the variance of Err_{XY} (which also has mean Err), showing that Err_X is **close** to Err.

- Note that in the example, MSE of $\hat{Err}^{(CV)}$ is **similar** when estimating either Err or Err_X, but **significantly different** when estimating Err_{XY}.



- **Bates, Hastie, and Tibshirani (2021)** show that $\hat{Err}^{(CV)}$ is **closer** to Err and Err_X than to Err_{XY} in the proportional asymptotic limit (for $n > p$, as $n, p \to \infty$ with $n/p \to \lambda > 1$).
- Combined with the earlier results, this implies that $\hat{Err}^{(CV)}$ is a **better estimator** for Err than for Err_{XY}.
- **Bates, Hastie, and Tibshirani (2021)** also show that $\hat{Err}^{(CV)}$ is **asymptotically uncorrelated** with Err_{XY}.

- Let $e_i = \ell\big(\hat{f}\,(x_i, \hat{\theta}^{(-1)}), y_i\big)$ be the **error** for each $i \in \mathcal{I}_k$ ($k = 1, \ldots, K$), resulting m different e_i's for each $\mathcal{I}_k$.
- Then, we can **re-define** **CV point estimate of prediction error** as the **average of errors** as follows:

$$\hat{Err}^{(CV)} := \frac{1}{K}\sum_{k=1}^{K}\frac{1}{m}\sum_{i\in\mathcal{I}_k}\ell\big(\hat{f}\,(x_i, \hat{\theta}^{(-1)}), y_i\big) = \frac{1}{n}\sum_{i=1}^{n} e_i = \bar{e},$$

- where $n = K \times m$.

- Assuming that e_i's are i.i.d., then **estimate of the standard error of CV point estimate of prediction error** would be:

$$\hat{se}^{(CV)} := \frac{1}{\sqrt{n}} \times \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} (e_i - \bar{e})^2},$$

- where the second term in the multiplication refers to the **empirical standard deviation** of the e_i.

A $100(1-\alpha)\%$ confidence interval for prediction error

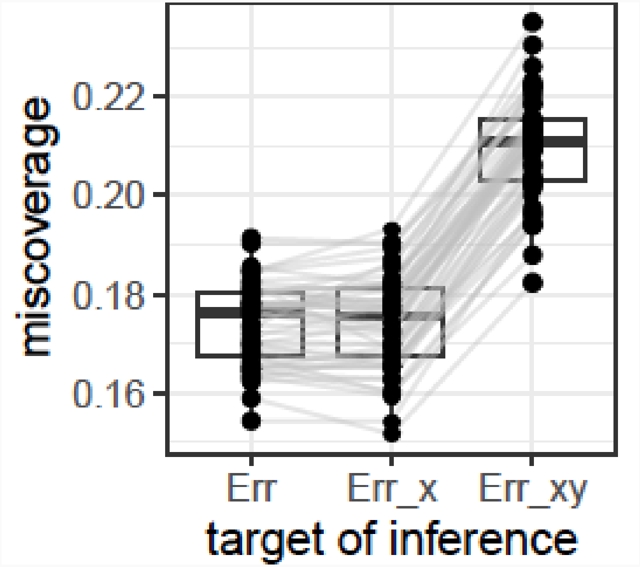
- A $100(1-\alpha)\%$ **confidence interval for prediction error** can be constructed as follows:

$$(\hat{Err}^{(CV)} - z_{1-(\frac{\alpha}{2})} \times \hat{se}^{(CV)} \quad , \quad \hat{Err}^{(CV)} + z_{1-(\frac{\alpha}{2})} \times \hat{se}^{(CV)}),$$

- where $0 < \alpha < 1$, $z_{1-(\frac{\alpha}{2})}$ is the $1-(\frac{\alpha}{2})$ quantile of the standard normal distribution.
- The intervals are called as **naive cross-validation intervals**.
- However, since every data point is used in both in training and testing, we **cannot accept** that are e_i's are **independent** of each other.
- Any **confidence interval** built on this assumption would have **poor coverage**.

Example re-visited

- The **naive cross-validation intervals** for **three estimands**: Err, Err_X, and Err_{XY} are built and miscoverage rates are estimated.

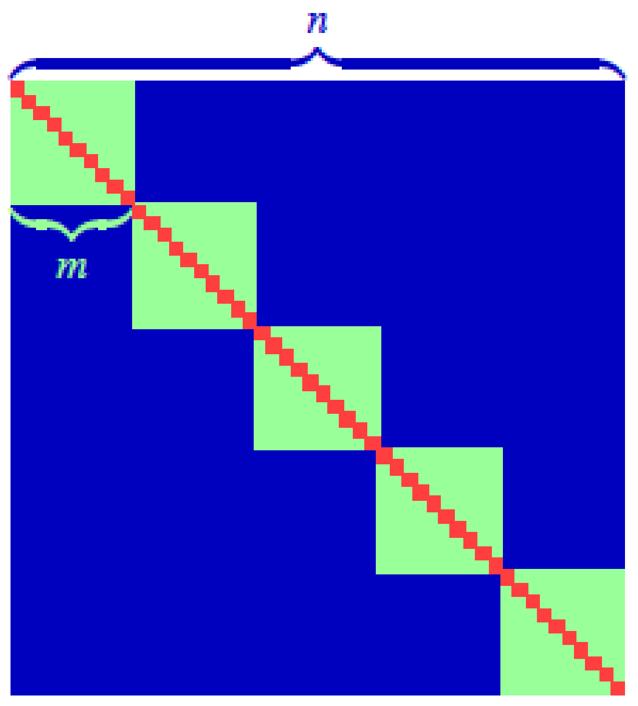


- The naive CV intervals **fail** due to large miscoverage rates.
- **Side note:** The nominal miscoverage rate is 10%.

- The fundamental paper of **Bengio and Grandvalet (2004)** gives the key structure of the covariance matrix of e_i's such that:

$$Var(\hat{Err}^{(CV)}) = \frac{1}{n}a_1 + \frac{n/K - 1}{n}a_2 + \frac{n - n/K}{n}a_3,$$

- where $a_1 = Var(e_i)$ is the **variance of the diagonal elements**,
- $a_2 = Cov(e_i, e_j)$ is the **covariance of the off-diagonal elements within the same fold** (in-block covariance of errors due to a common training set), and
- $a_3 = Cov(e_i, e_j)$ is the **covariance between-blocks**, covariance due the dependence between training sets $\mathcal{I}_k$ $(k = 1, \ldots, K)$.



Structure of the covariance matrix of errrors.

Image Source

- The constants a_2 and a_3 will typically be positive, in which case:

$$Var(\hat{Err}^{(CV)}) > \frac{1}{n}a_1,$$

- The naive cross-validation intervals **implicitly assume** $a_2 = 0$ and $a_3 = 0$.
- Hence estimating the variance of $\hat{Err}^{(CV)}$ as $\hat{se}^2$ results in an estimate that it is too **small**, and, in turn, **poor coverage**.

Target of inference: Confidence intervals

- **Bates, Hastie, and Tibshirani (2021)** develops an **estimator** that **empirically estimate the variance** of $\hat{Err}^{(CV)}$ across many subsamples.

- **Definition 2:** For a sample of size n split into K folds, the cross-validation MSE is:

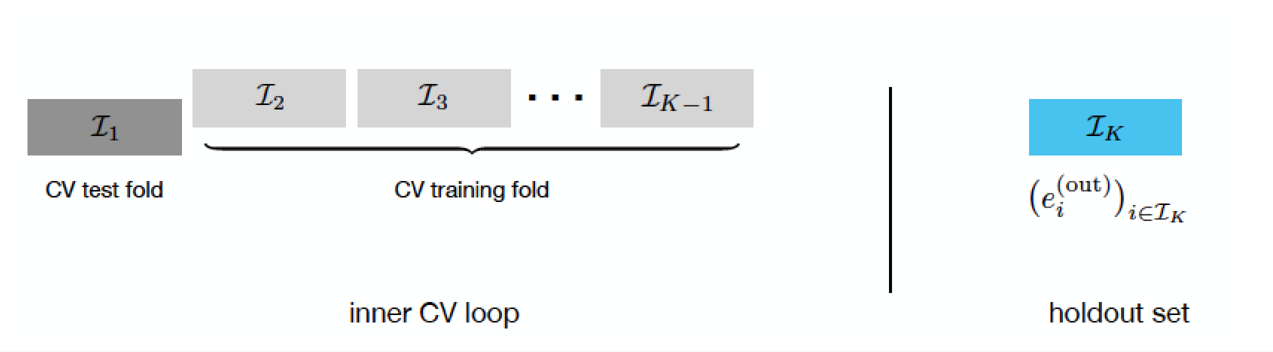
$$MSE_{K,n} := \mathbb{E}\left[(\hat{Err}^{(CV)} - Err_{XY})^2\right].$$

- MSE contains both a **bias term** and **variance term**, but the **bias** typically **small** for cross-validation (Efron, 1983; Efron and Gong, 1983; Efron and Tibshirani, 1997).
- **MSE** can be viewed as a slightly conservative version of the **variance** of the cross-validation estimator.
- The estimate of the MSE can be used to construct confidence intervals for Err_{XY} since it is what typically matters for practical applications.

- **Lemma 2:** For a single split, randomly partition the data into a training set with $K-1$ folds and denote it by $\mathcal{I}_{(train)} = \cup_{k=1}^{K-1}\mathcal{I}_k = (\tilde{X}, \tilde{Y})$, and call the remaining fold as $\mathcal{I}_{(out)}$. Using only $(\tilde{X}, \tilde{Y})$, define the prediction error $Err_{\tilde{X},\tilde{Y}}$ and an estimator $\widehat{Err}_{\tilde{X},\tilde{Y}}$ such as cross-validation, as usual. For the hold-out data set, calculate errors $\{e^{(out)}\}_{i\in I_{(out)}}$ and their average $\bar{e}^{(out)}$.

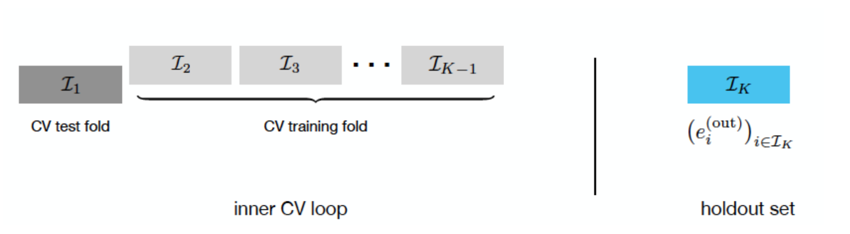
- Then, estimate the MSE from the data as follows:

$$\mathbb{E}\Big[\big(\hat{Err}_{\tilde{X},\tilde{Y}} - Err_{\tilde{X},\tilde{Y}}\big)^2\Big] = \mathbb{E}\Big[\big(\hat{Err}_{\tilde{X},\tilde{Y}} - \bar{e}^{(out)}\big)^2\Big] - \mathbb{E}\Big[\big(\bar{e}^{(out)} - Err_{\tilde{X},\tilde{Y}}\big)^2\Big].$$



Nested CV algorithm

- **Repeatedly** split data into K folds with $K-1$ building $\mathcal{I}_{(train)}$ and the remaining one being $\mathcal{I}_{(out)}$.
- For each split j:
  - Compute $\epsilon_j := \hat{E}rr_{\tilde{X}\tilde{Y}}$ with (K-2)-fold cross-validation over K-1 folds in $\mathcal{I}_{(train)}$.
  - Train model on $\mathcal{I}_{(train)}$ and compute errors e_i for all data points $(x_i, y_i) \in \mathcal{I}_{(out)}$.
  - Compute $\bar{e}_{out}$:= mean of $\{i\}_{e \in I_{out}}$.
  - Set $a_j := (\hat{E}rr_{\tilde{X}\tilde{Y}} - \bar{e}_{out})^2$ (estimate of (the first term at RHS)).
  - Set b_j := empirical variance of $\{e_i\}_{i \in I_{out}}$ (estimate of (the second term at RHS)).
- Output $\hat{MSE} = mean(a_j) - mean(b_j)$.
- Output $\hat{E}rr^{(NCV)} := mean(\epsilon_j)$.



- Nested CV algorithm provides us a **point estimate** for prediction error, denoted by $\hat{Err}^{(NCV)}$, and **estimate for MSE**, denoted by $\hat{M}SE$.

- **Theorem 2:** For a nested CV with a sample of size n:

$$\mathbb{E}\left[\hat{M}SE\right] := MSE_{K-1,((K-1)n/K)},$$

- where n/K is the fold size.
- Since the estimation is done over $K-1$ folds, $\hat{M}SE$ estimates the actual quantity of interest $MSE_{K,n}$ with some **bias**.
- $\hat{M}SE$ is **rescaled** to obtained unbiased estimate for $MSE_{K,n}$.
- Similarly, $\hat{Err}^{(NCV)}$ is also adjusted (**de-biased**).

A $100(1-\alpha)\%$ confidence interval

- Finally, a $100(1-\alpha)\%$ confidence interval is obtained as follows:

$$(\hat{Err}^{(NCV)} - \hat{bias} - z_{1-(\frac{\alpha}{2})}\hat{se}^{(NCV)}, \quad \hat{Err}^{(NCV)} + \hat{bias} - z_{1-(\frac{\alpha}{2})}\hat{se}^{(NCV)}),$$

- where

- $$\hat{bias} := (1 + (\frac{K-2}{K}))(\hat{Err}^{(NCV)} - \hat{Err}^{(CV)}) \quad \text{and} \quad \hat{se}^{(NCV)} := \sqrt{\frac{K-1}{K}}\sqrt{\hat{MSE}}.$$

Simulation experiments: Data generation scenario

- The **coverage** of nested CV intervals approach is investigated for **classification** and **regression** problems over synthetic data sets (and real data sets).
- Consider a **sparse logistic data generating model**:

$$Pr(Y_i = 1 | X_i = x_i) = \frac{1}{1 + \exp(-x_i^T \theta)}, \quad \text{i=1,...,n,}$$

- where n is the number of observations, p is the number of features,
- X_i is the feature matrix consisting of i.i.d standard Gaussian variables,
- the coefficient $\theta = c \times (1, 1, 1, 1, 0, 0, \dots)^T \in \mathcal{R}^p$ and
- c is chosen such that **Bayes error** is either 33.2% or 22.5% which is the **optimal lower bound** for Err.

Simulation experiments: Performance metrics

- In each case:
  - The **miscoverage** of naive CV (CV) and nested CV (NCV) intervals are reported where the **nominal miscoverage** rate is 10%.
  - The **width** of NCV intervals are expressed relative to the width of CV intervals.
  - 10-fold CV and 10-fold-NCV with 200 splits are used.
- R scripts of reproducing experiments are available at: https://github.com/stephenbates19/nestedcv_experiments.

Simulation results: Low-dimensional setting results

- $n = 100, p = 20$, and (un-regularized) logistic regression is used as fitting algorithm.
- Nested CV gives **coverage closer** to the nominal level.

Bayes Error	Target	Width	Point Estimates		Miscoverage			
					CV		NCV	
		NCV	CV	NCV	Hi	Lo	Hi	Lo
33.2%	Err_{XY}	1.23	39.6%	39.0%	10%	8%	3%	5%
"	Err	"	"	"	9%	8%	3%	4%
22.5%	Err_{XY}	1.47	30.4%	28.1%	11%	3%	4%	1%
"	Err	"	"	"	10%	2%	5%	0%

- A "Hi" miscoverage is one where the **confidence interval is too large** and the point estimate falls below the interval; conversely for a "Lo" miscoverage.

Simulation results: High-dimensional setting results

- $n = 90, 200, p = 100$, and ℓ_1 penalized logistic regression with a fixed penalty level is used as fitting algorithm.
- Nested CV gives **coverage more closer** to the nominal level.

n	ρ	Bayes Error	Target	Width NCV	Point Estimates CV	Point Estimates NCV	Miscoverage CV Hi	Miscoverage CV Lo	Miscoverage NCV Hi	Miscoverage NCV Lo
90	0	22%	Err_{XY}	1.53	41.8%	41.1%	16%	12%	6%	7%
90	0	22%	Err	1.53	41.8%	41.1%	17%	13%	6%	8%
200	0	22%	Err_{XY}	1.66	26.7%	25.6%	14%	7%	3%	5%
200	0	22%	Err	1.66	26.7%	25.6%	15%	7%	4%	6%
90	0.5	13%	Err_{XY}	1.80	27.5%	28.6%	20%	10%	5%	8%
90	0.5	13%	Err	1.80	27.5%	28.6%	20%	11%	7%	9%

Discussion & Future work

- Nested CV is more **computationally intensive** than standard CV, but, parallel programming can be used for speeding up the algorithm.
- It is well-known that CV is also commonly used for selecting a good value of a learning algorithm's hyperparameters (fine-tuning).
- **Bates, Hastie, and Tibshirani (2021)** expect that NCV would be of use for hyperparameter selection since it yields more accurate confidence intervals for prediction error.

Merci!..